Next Generation Infrastructure for Scalable Displays

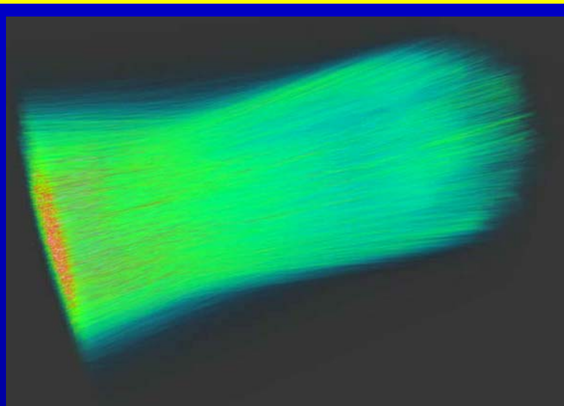
Philip Heermann & Randall Frank

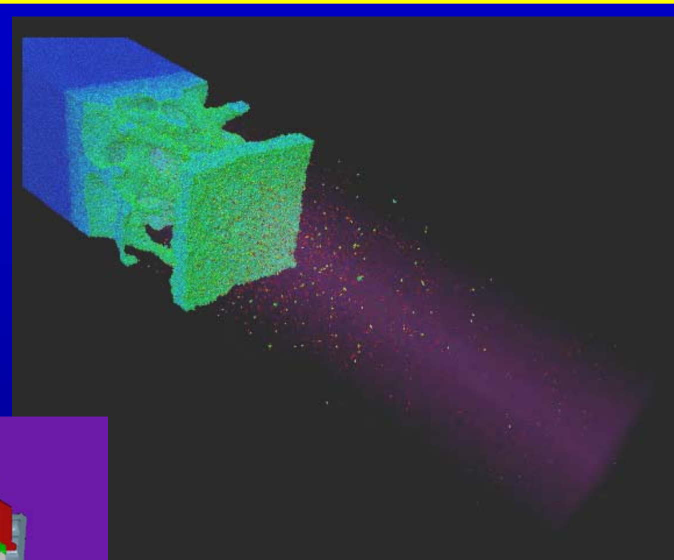
Sandia National Laboratories

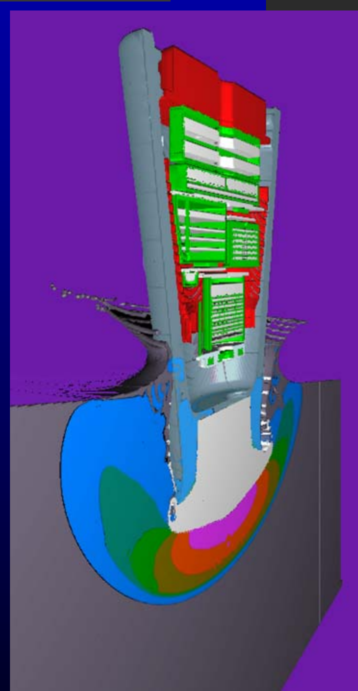
Lawrence Livermore National Laboratory

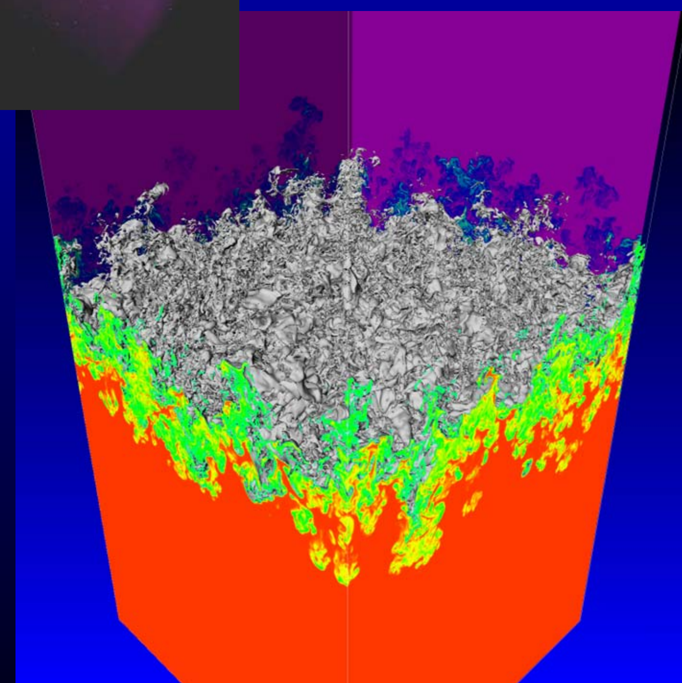
UCRL-PRES-200237

Examples of Scientific Data of Interest





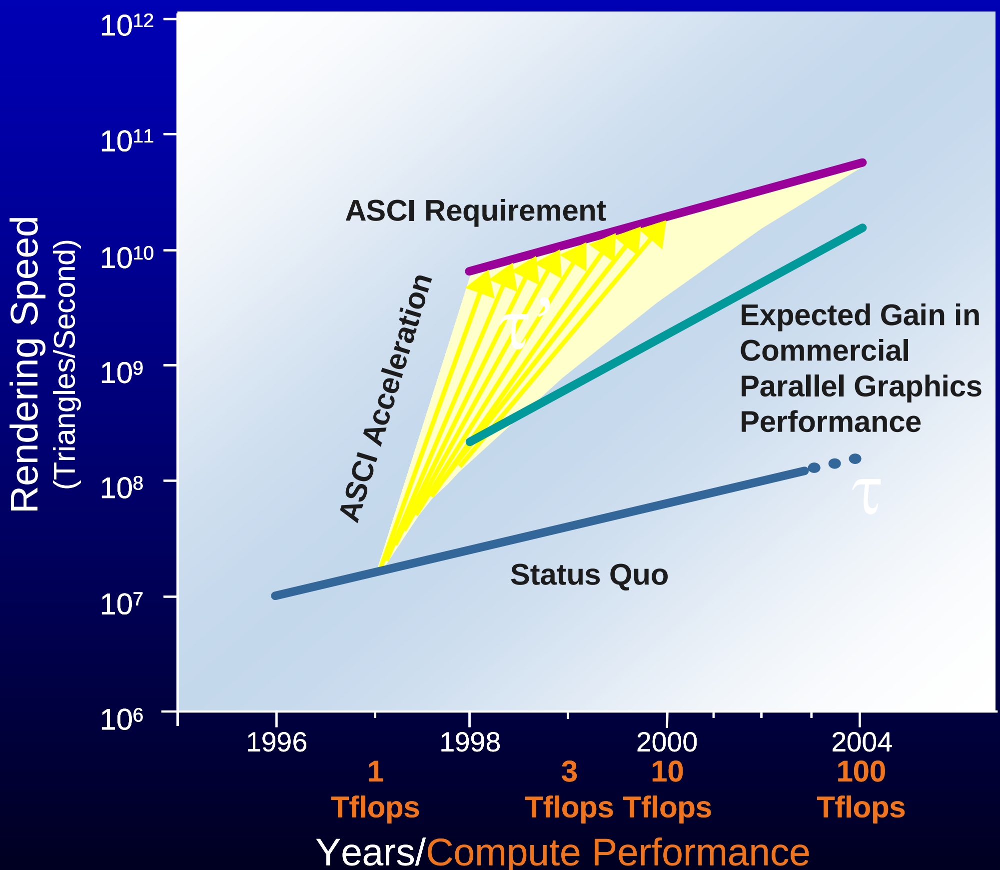




Motivation for Scalable Visualization

Production Delivery of
Technology for Dept of Energy

- ASCI Requirement 1000X increase in Rendering 1998-2004
- Concurrent with Disruptive Technology Transition



Factors governing ASCI Visualization

- **The realities of extreme dataset sizes**
  - Stored with the compute platform
  - Cannot afford to copy the data
  - Visualization co-resident with compute platform
- **Track compute platform trends**
  - Distributed infrastructure
  - Commodity hardware trends
- **Migration of graphics leadership to the PC**
  - In clusters, desktops and displays...

Scientific Visualization: Data to Display

Data Sources: Simulations, Archives, Experiments, Sensors

Data Services:
- Permutation M X N
- Filtering
- 1D/2D Subsetting
- Data Algebra x,y,z to mag/phase
- Format/Representation Conversion
- Data Reduction
- Data Serving

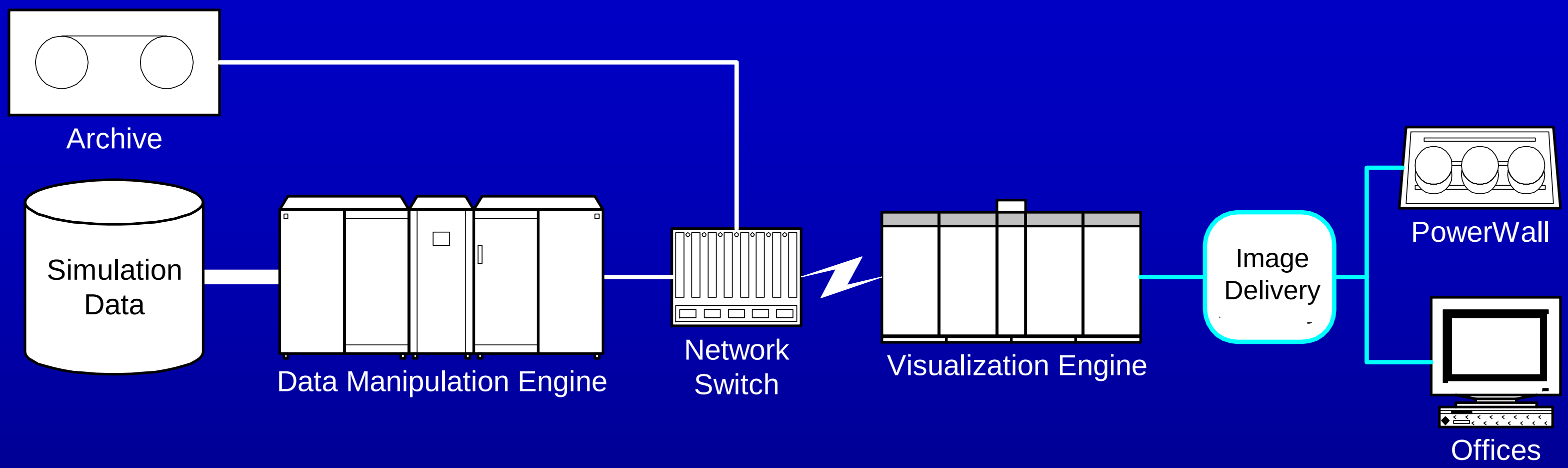
Information Services:
- Feature Detection and Extraction
- Data Fusion & Comparison
- Visual Representations Generation (eg. isosurfaces)
- Volume Visualization Preparation (eg. opacity assignment, resampling ...)

Visualization Services:
- Surface rendering
- Volume rendering
- Runtime services
- Multi-Visualization Technique Combine
- Time Sequence Generation

Display Modalities:
- Desktop Display
- Theater Display
- Powerwalls
- Immersive Stereoscopic

Users Services:
- Navigation
- Rendering Control
- Advanced User Interface
- Collaborative Control
- Display Control

Tri-lab Model for ASCI Visualization



- Raw data on platform disks/archive systems
- Data manipulation engine (direct access to raw data)
- Networking (data/primitives/images)
- Visualization/rendering engine
- Video and remotely rendered image delivery over distance
- Displays (office/PowerWalls)

Major ASCI Visualization Focus

High-performance access to very large data sets
- parallel data streams from source to eye

High Resolution Displays

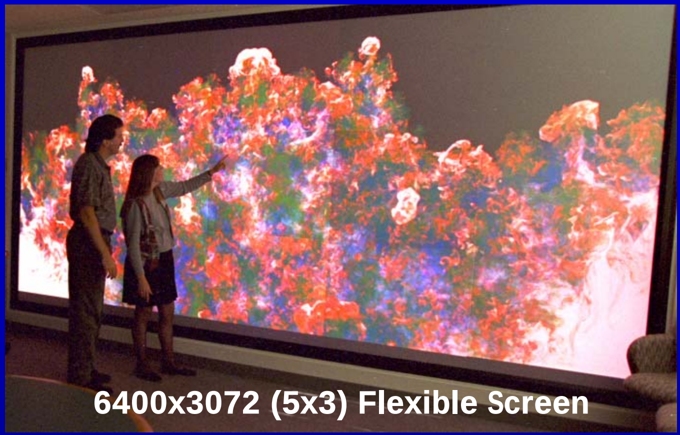
- Shared Facilities
- Desktops

Scalable Visualization

- Scalable Rendering
- Scalable Data Handling

The Tiled Display PowerWall

- **Many forms…**
  - Stereo, Cubes, Front/Back projection, Non-planar, Edge-blended, CAVEs
- **Multiple uses**
  - Collaborative environments, Theaters, Enhanced desktop/interactive use
- **Increased pixel counts**
  - Matching higher fidelity data
- **Driving a PowerWall**
  - Extreme I/O requirements
    - 2x2 needed 300MB/s
  - Synchronization (at a distance?)
  - Data flow and data staging
  - Requires output scalable image generation via aggregation



6400x3072 (5x3) Flexible Screen

5120x2048 (4x2) Glass Screen

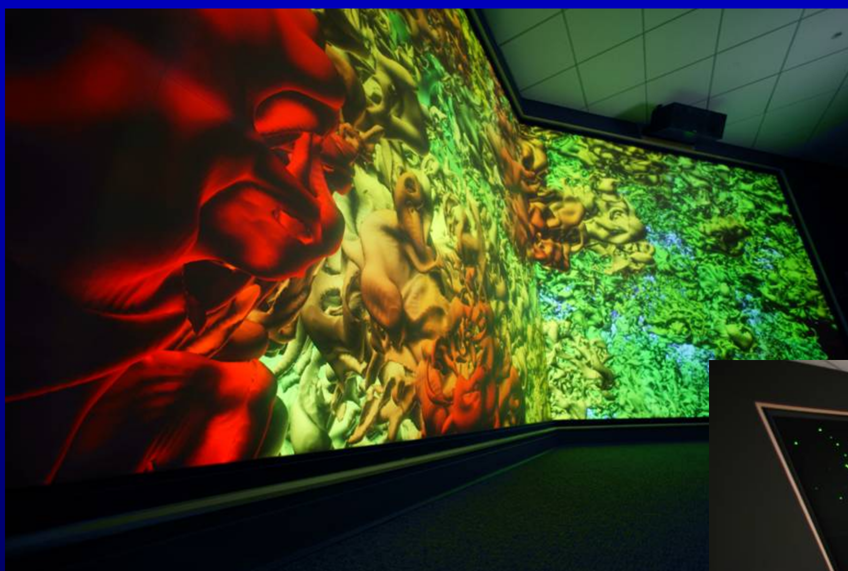
3840x2048 (3x2) Cubes

Examples: Systems in use Today...

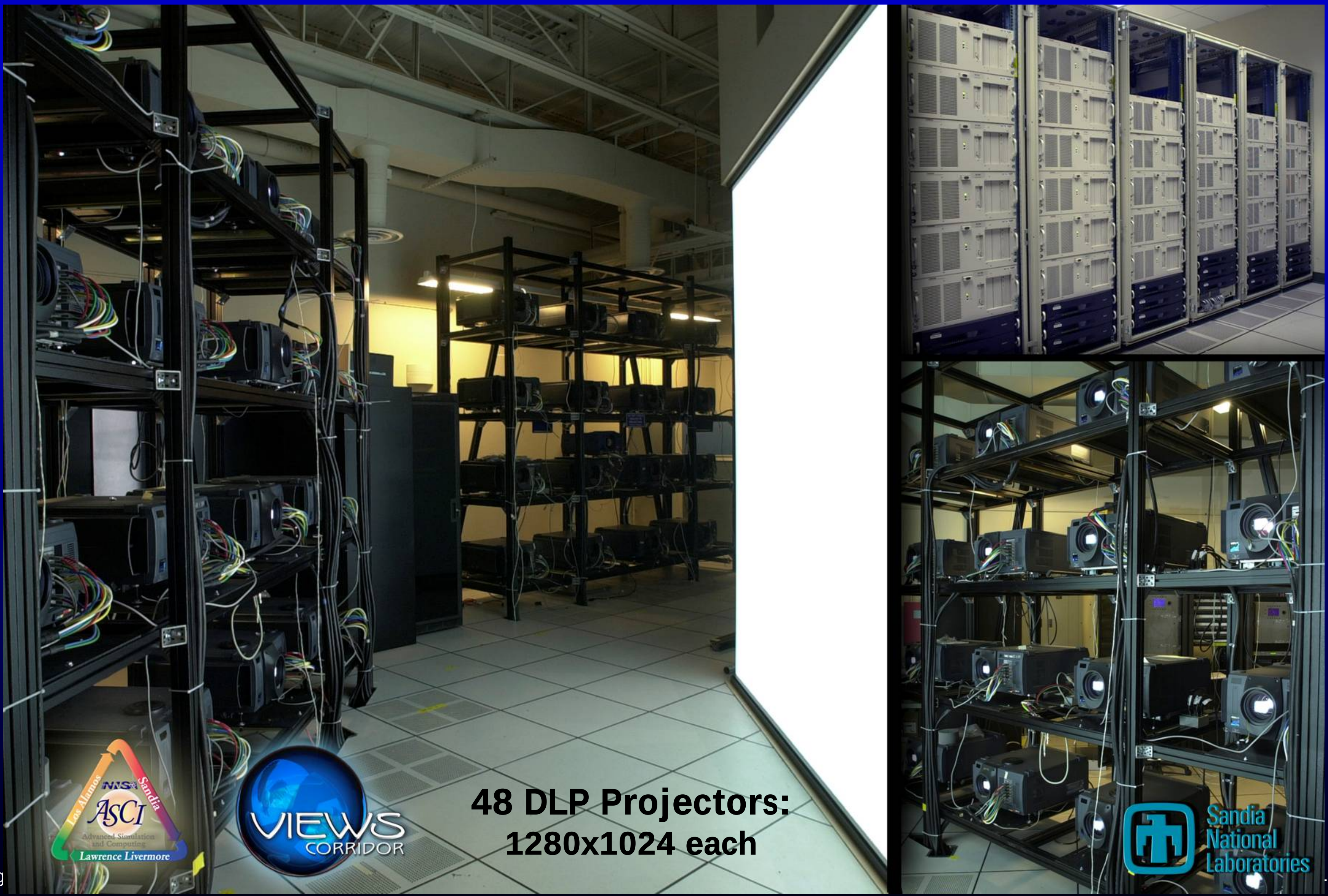
- SGE: digitally driven displays
- SNL 62Mpixel wall surfaces
- DMX: desktop integration

Large Scale Visual Acuity Display

Large Scale Visual Acuity Display



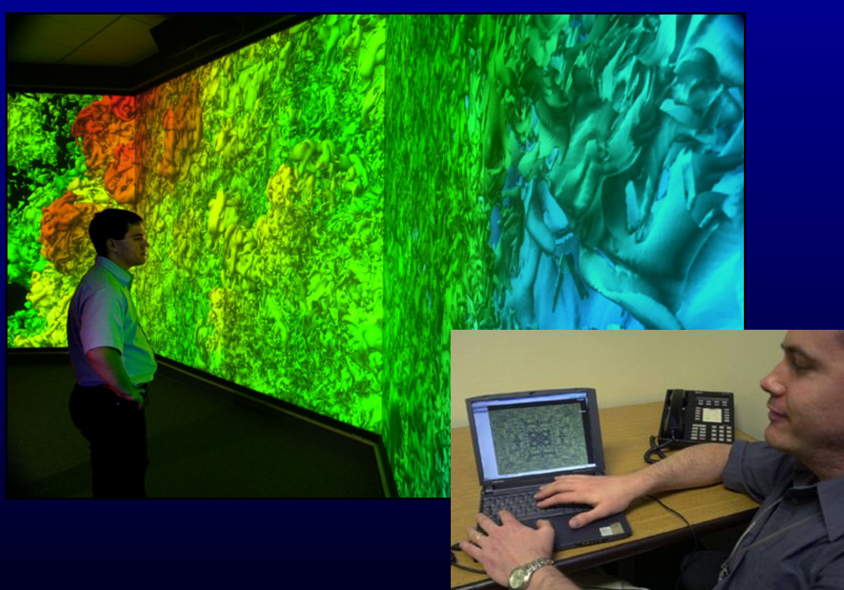
Rendering and Display System



Scalable Rendering & Display

- 62 MegaPixel Resolution Display
- 1 Billion Poly/S (to single display)
- 60 Million Poly/S (to 62MP display)

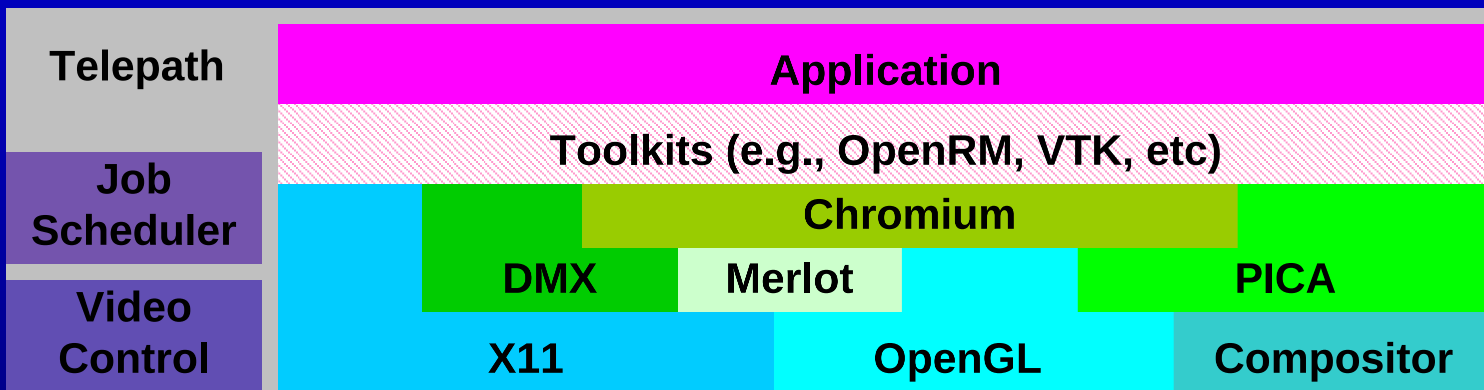
Major Progress toward replacing SMP Visualization Servers



Video

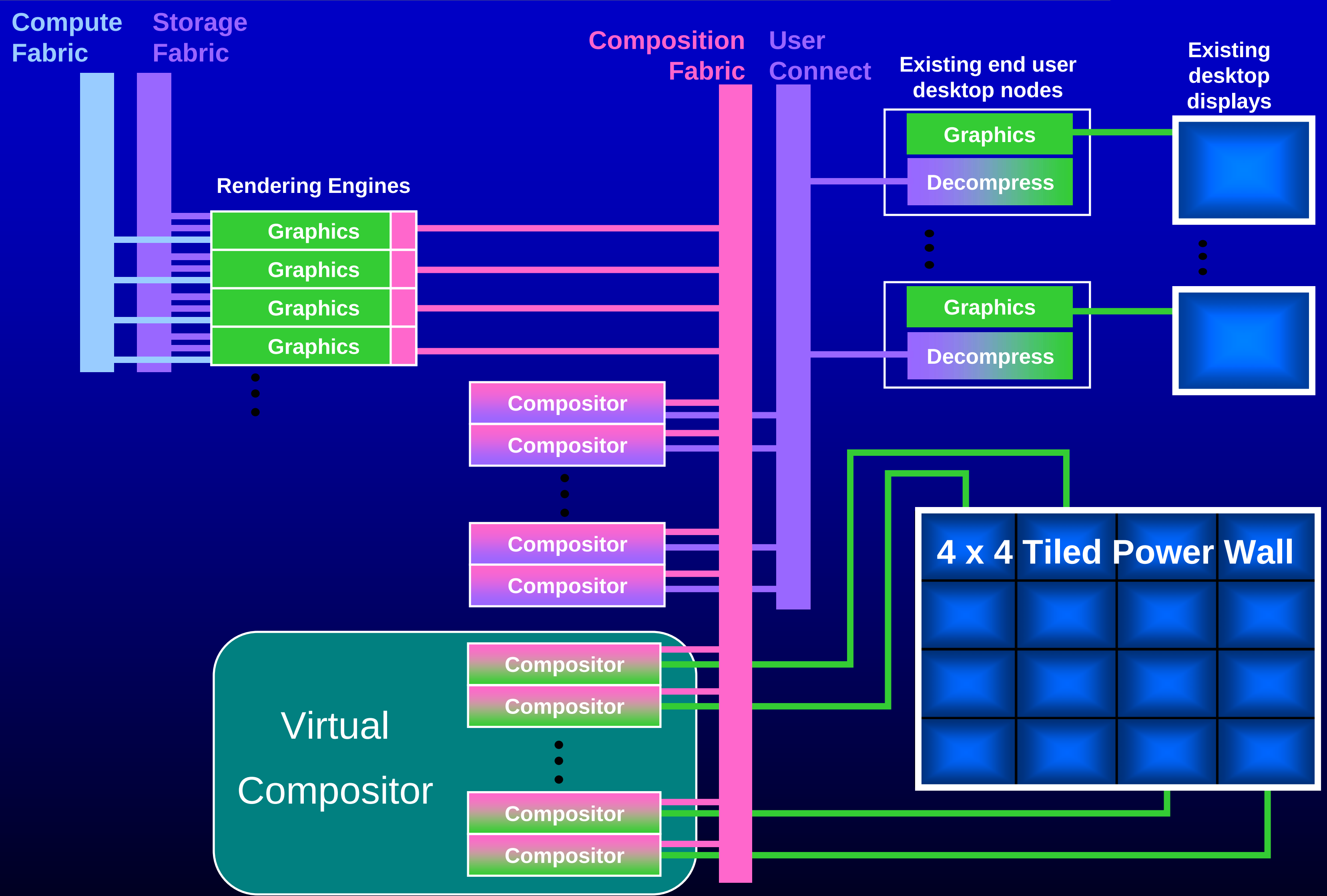
Robust Scalable Visualization: Software Tools

Goal: Provide integrated, distributed parallel services for viz apps. Encourage new apps, increase portability & device transparency.

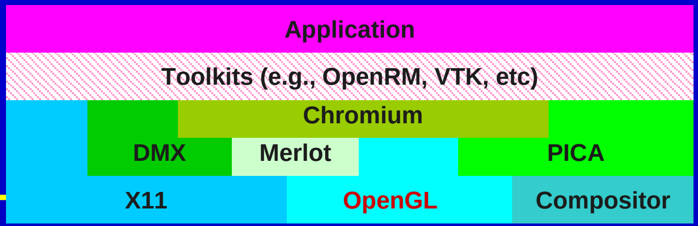


- Applications: VisIt, ParaView, EnSight, Blockbuster, etc.
- Toolkits – Parallel visualization algorithms, scene graphs, etc.
- DMX – Distributed X11 windowing and input devices
- Chromium – Parallel OpenGL rendering model
- PICA – Parallel image compositing API
- Merlot – Digital image delivery infrastructure
- Telepath – Visualization “session” control and scheduling
- Core “vendor” services - X11/OpenGL/compositors/NICs/Job control

Idealized Visualization Environment

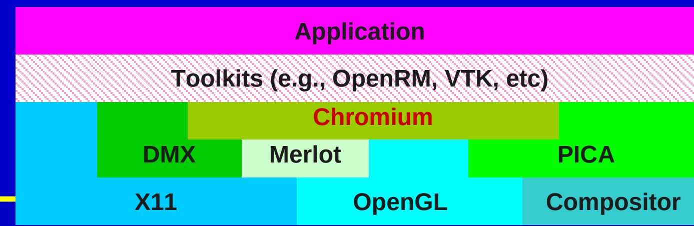


OpenGL Drivers

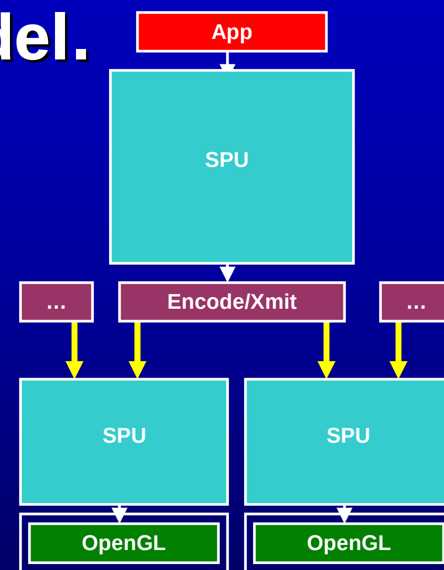


- **COTS/Linux OpenGL drivers are looking good**
  - Solid support from nVidia, ATI and others
  - Drivers support recent ARB extensions
    - Vertex & fragment programs, ARB_vertex_buffer_object, etc
    - Complete buffer support: “float” pbuffers, multi-head, stereo, etc
  - Excellent core performance, but increasing in somewhat tangential directions
- **A very dynamic situation**
  - API “richness” often results in buggy implementations
  - Welcome to the world of extensions…

Distributed GL: Chromium



Distributed OpenGL rendering pipeline. Provides a parallel OpenGL interface for an N to M rendering infrastructure based on a graphics stream processing model.



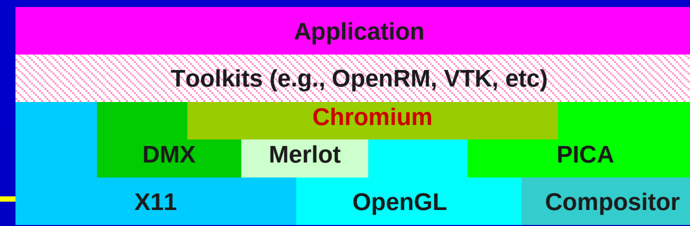
The Stream Processing Unit (SPU)

- “Filter” view OpenGL
- SPU interface is the OpenGL API
  - Render, modify, absorb…
- Allows direct OpenGL rendering
- Supports SPU inheritance
- Application “translucent”

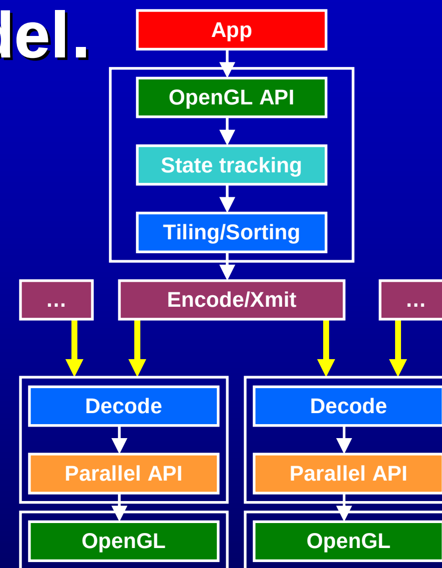
Development:

- chromium.sourceforge.net
- RedHat/Tungsten Graphics ASCI PathForward
- Stanford, University of Virginia
- Stereo, Fragment/Vertex pgms, CRUT, dynamic caching

Distributed GL: Chromium



Distributed OpenGL rendering pipeline. Provides a parallel OpenGL interface for an N to M rendering infrastructure based on a graphics stream processing model.



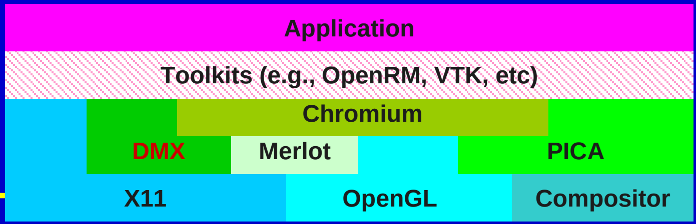
The Stream Processing Unit (SPU)

- "Filter" view OpenGL
- SPU interface is the OpenGL API
  - Render, modify, absorb…
- Allows direct OpenGL rendering
- Supports SPU inheritance
- Application "translucent"

Development:

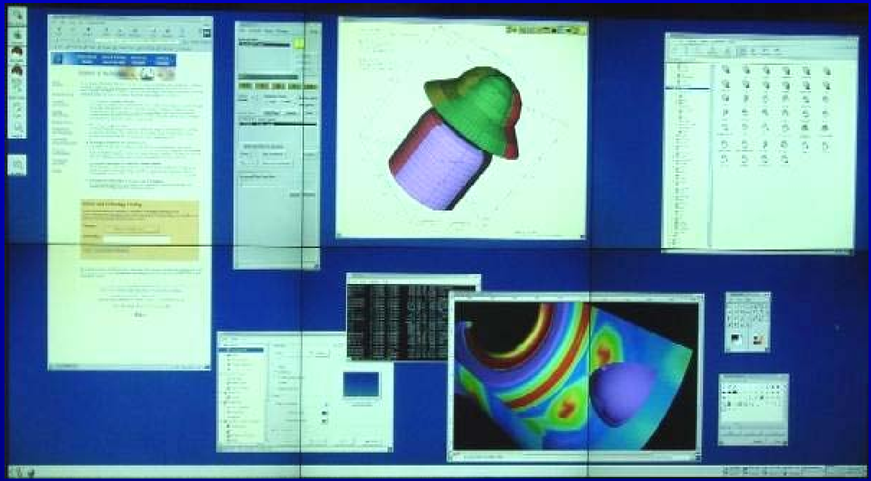
- chromium.sourceforge.net
- RedHat/Tungsten Graphics ASCI PathForward
- Stanford, University of Virginia
- Stereo, Fragment/Vertex pgms, CRUT, dynamic caching

Parallel X11 Server: DMX



Distributed multi-headed X server: DMX

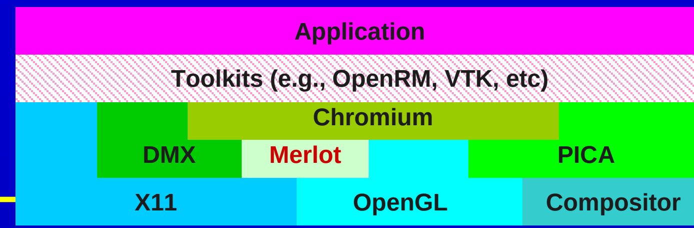
- **Aggregates X11 servers**
  - "Server of servers" for X11
  - Single X server interface
- **Accelerated graphics**
  - 2D via accelerated X server
    - Common extensions as well
  - Back-side APIs for direct, local X11 server access
  - OpenGL via ProxyGL/GLX (from SGI) or via Chromium SPU



Development:

- dmx.sourceforge.net
- RedHat ASCI PathForward contract
- Integrated with XFree86

Remote Delivery: Merlot

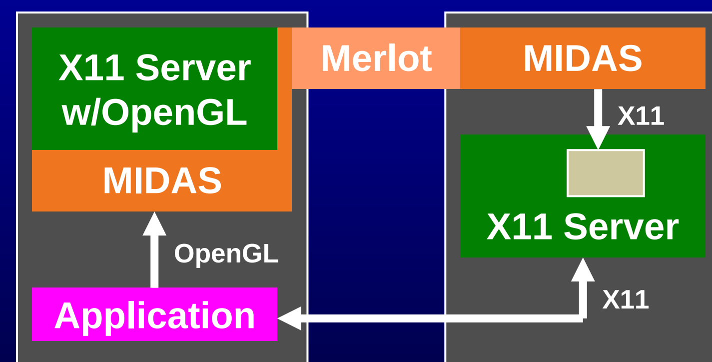


Merlot is a framework for digital image delivery

- Transport layer abstraction, Codec interfaces, Device transparency

MIDAS: Merlot Image Delivery Application Service

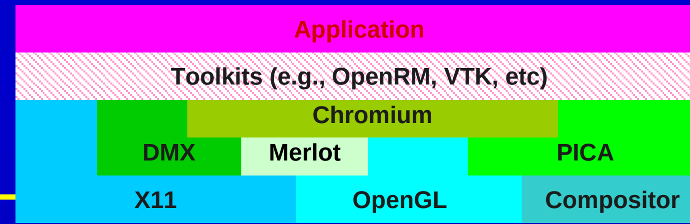
- Indirect OpenGL rendering services for X11 environment
- Indirect window management
- Image stream transport



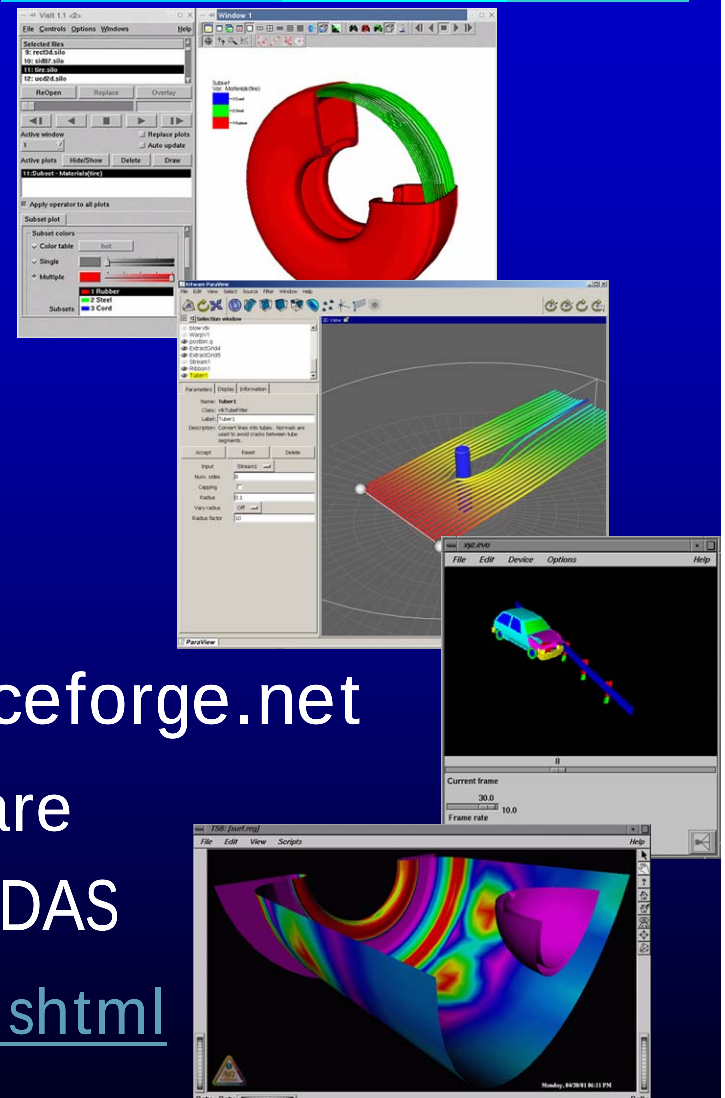
Development:

- Alpha released as OpenSource (on SourceForge?)
- More apps and experimental hardware support

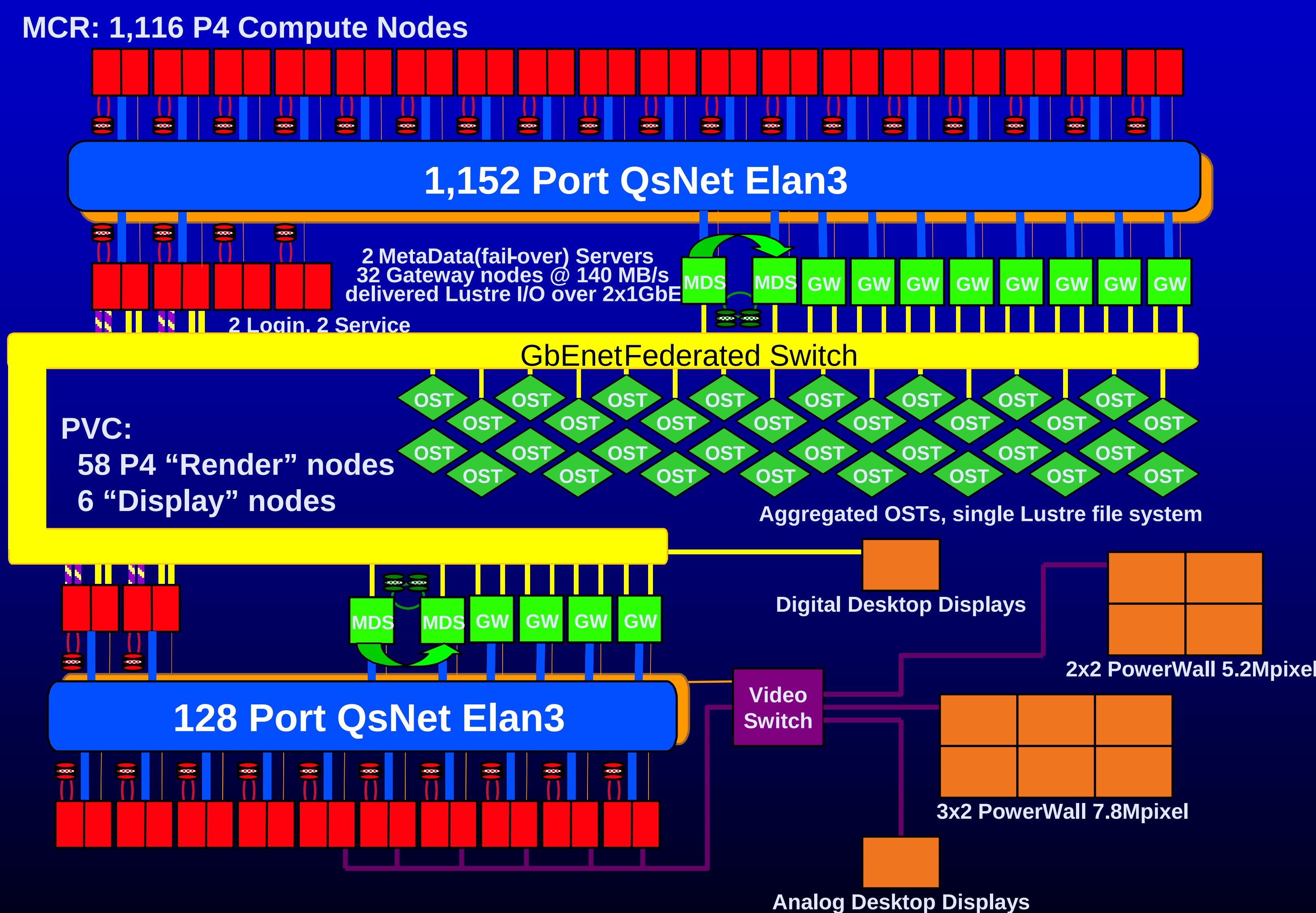
Applications



- **Full-featured visualization**
  - VisIt: www.llnl.gov/visit
    - VTK, client-server model
  - ParaView: www.paraview.org
    - Parallel VTK viz tool
- **Specialty applications**
  - Blockbuster: blockbuster.sourceforge.net
    - Scalable animations, DMX aware
  - TeraScale Browser/Xmovie/MIDAS
    - www.llnl.gov/icc/sdd/img/viz.shtml



Deployed Environment: MCR & PVC

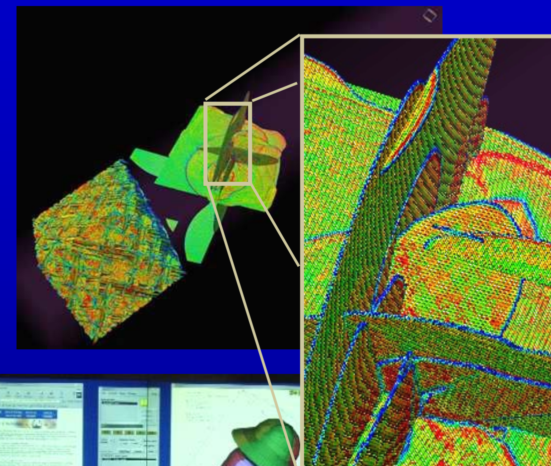


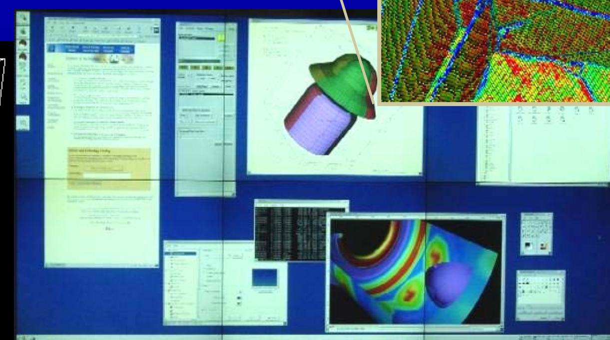
Current MCR & PVC Status

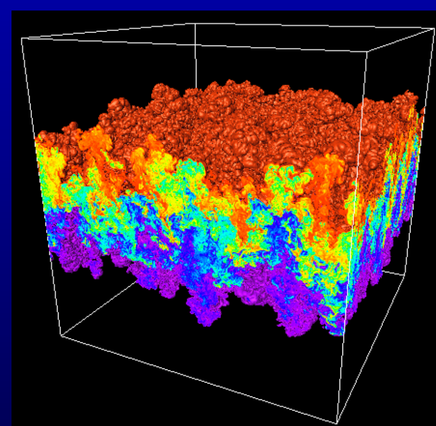
- **Early GA, but it has been extremely productive**
  - Easily handling multi-TB data and multi-Mpixel displays
  - Generating animations in 1/10 the time using 1/5 the resources
- **Challenges remain…**
  - Data access
    - Cluster-wide parallel I/O systems can be fragile
    - Often not optimized for viz access patterns (e.g. random reads)
  - The impedance mismatch problem
    - Smaller number of nodes generally for visualization
    - Improper data decomposition
  - Scheduling complexity
    - Co-scheduling of multiple clusters
    - Combinations of parallel clients, servers, services and displays

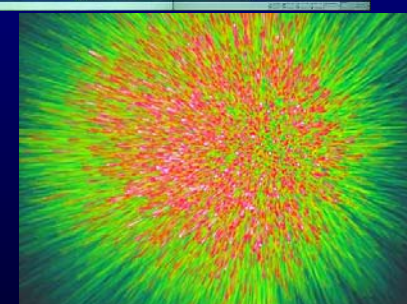
Data Challenges: Large & Complex

- **Increased Complexity**
  - Increased fidelity
    - Spatial, temporal, higher-order
  - Complex & unique representations
- **PSE Integration**
  - Multiple data sources
  - Additional context
- **Algorithmic failure**
  - Difficult interpretation (e.g. depth complexity)
    - Increased use of stereo display
  - Scalability: global algorithms and rendering fidelity









The Road Ahead

- **Longevity challenge: software and hardware**
  - The VIEWS Open Source software stack
  - New graphics bus (PCI Express)
  - Next generation graphics cards
- **On the horizon...**
  - New rendering abstractions (are polygons dead?)
    - How to address current card bottlenecks (e.g. setup)
  - Changes in video technologies
    - DVI to 10gigE (TeraBurst)
  - Dedicated compositing hardware
  - The extreme FLOP approach

Image Aggregation Solutions

Image "compositing": Take the (digital) outputs of multiple graphics cards and combine them to form a single image.

Multiple goals/dimensions of scaling via aggregation

- Output scaling: Large pixel counts (PowerWalls)
- Data scaling: High polygon/fill rates/data decomposition
- Interaction/Virtual reality: High frame rates
- Image quality: Anti-aliasing, data extremes

Hardware acceleration is natural

- Efficient access to rendered imagery
- Provide for image "fragment" transport
- Flexible, pipelined "merging" operations

Solutions balance speed, scale…

- Image input/transport solutions
- Application transparency
- Parallel rendering models

HP sv6 & sv7

Examples: Compositing Systems

Image composition hardware

- **Lightning-2/MetaBuffer (Stanford/UT)**
- **sv6/sv7 (HP)/SGI compositor**
- **ORAD compositor**
  - DVI based tiling/compositing
- **Sepia (HP/Compaq)**
  - Custom compositing (FPGA + NIC)
  - Dedicated network (ServerNet II & IB)
- **Scalable Graphics Engine (IBM)**
  - Remote framebuffer, gigE/UDP distance solution

Lightning-2

HP Sepia-2

IBM SGE

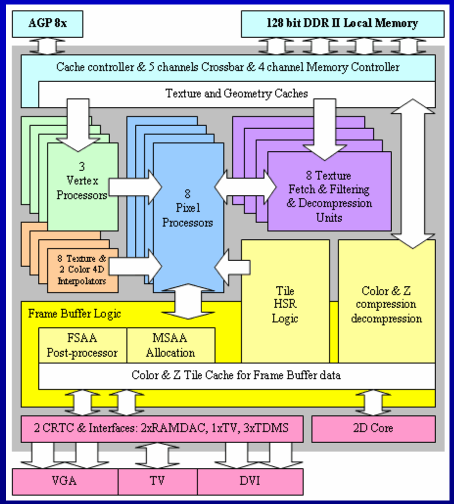
Image composition software

- PICA: Parallel Image Compositing API
- ICE-T: Integrated image/data manipulation (SNL)

The NV30 and the Sony Playstation 3

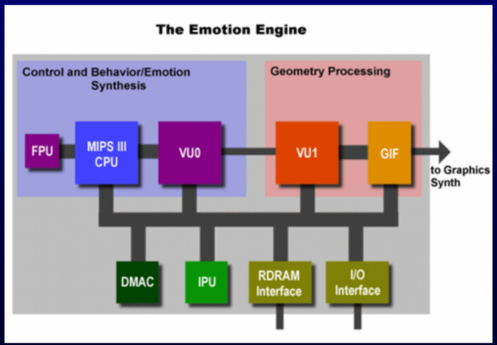
Are graphics trends a glimpse of the future?

- **The nVidia NV30 Architecture**
  - 256MB+ RAM, 96 32bit IEEE FP units @ 500Mhz
  - “Assembly language” for custom operations
  - Streaming internal infrastructure



nVidia NV30

- **The PlayStation3 (patent application)**
  - Core component is a cell
    - 1 “PowerPC” CPU + 8 APUs (“vectorial” processors)
    - 4GHz, 128K RAM, 256GFLOP/cell
    - Building block for multimedia framework
  - Multiple cells
    - Four cell architecture (1TFLOP)
    - Central 64MB memory
    - Switched 1024 bit bus, optical links?



Sony PS2 “Emotion” Engine

The Streaming Programming model

Streaming exposes concurrency and latency at the system level as part of the programming target

Data moves through the system: exposed concurrency

- Avoid global communication: prefer implicit models (e.g. Cr)

Memory model: exposed latency/bandwidth

- Scalable, must support very small footprints
- Distributed, implicit flow between each operation

A working model:

- Computational elements + caching and bandwidth constraints
- External “oracle” for system characterization and realization

Goals:

- Optimally trade off computation for critical bandwidth
- Leverage traditionally “hidden” programmable elements

Next Generation Streaming Cluster…

- **Computation and memory caches everywhere**
  - NICs, Drive controllers, Switches, TFLOP GPUs
  - Add PCI Express and the GPU effectively becomes a DSP chip
  - Utilizing them may require a disruptive programming shift
- **Modified visualization algorithms**
  - Cache oblivious: local, at the expense of computation
  - "Non-graphical" algorithms, moving away from polygon primitives
  - Need to address data scaling and representation issues
  - Digital, high dynamic range imagery from generation to display
- **New languages with higher levels of abstractions**
  - Run-time "realization", dynamic compilation and scheduling
  - Glue languages: "shader" languages, graphics APIs themselves

Auspices:

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48 and was partially supported by Sandia National Laboratories. Sandia is a multi-program laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.